

Emilio Gomez Breschi

19300100

4D1

Sonia Erika Ibáñez de la Torre

Desarrollo de Software

10/Octubre/2021

Programación Orientada a Objetos

***¿Que es un constructor y para que es utilizado?***

En programación, los constructores son funciones o métodos que permiten realizar tareas de instanciación de objetos. Cuando un objeto es creado a partir de una clase, se llama al constructor que se encargará de inicializar los atributos del objeto, como así también cualquier llevar a cabo cualquier otra función necesaria.

***Sintaxis del constructor , en la clase y cuando se saca de la clase***.

Class FooClass

{

public:

FooClass(/\*Parámetros de entrada opcionales \*/); //Constructor };

FooClass::FooClass()

{

//Líneas de código del constructor

}

class Box {

public:

// Default constructor

Box() {}

// Initialize a Box with equal dimensions (i.e. a cube)

explicit Box(int i) : m\_width(i), m\_length(i), m\_height(i) // member init list

{}

// Initialize a Box with custom dimensions

Box(int width, int length, int height)

: m\_width(width), m\_length(length), m\_height(height)

{}

int Volume() { return m\_width \* m\_length \* m\_height; }

private:

// Will have value of 0 when default constructor is called.

// If we didn't zero-init here, default constructor would

// leave them uninitialized with garbage values.

int m\_width{ 0 };

int m\_length{ 0 };

int m\_height{ 0 };

};

***¿Cuantos constructores se pueden tener en una clase y bajo que condiciones?***

Puede definir tantos constructores sobrecargados como sea necesario para personalizar la inicialización de varias maneras. Normalmente, los constructores tienen accesibilidad pública para que el código fuera de la definición de clase o la jerarquía de herencia pueda crear objetos de la clase . Pero también puede declarar un constructor como protected o private .

***Que es la ambigüedad en la sobrecarga de constructores***

Un int se puede convertir tanto en double como en unsigned; ambas son consideradas conversiones de "ampliación" y tienen el mismo rango. Como norma general , cada vez que sobrecarga y una de las sobrecargas es de tipo integral , es una buena idea sobrecargar también en int, para asegurarse de que obtenga lo que desea, y evitar ambigüedades, cuando alguien intenta a pasa una constante integral.

private:

t m\_Value;

unsigned int m\_Length;

***Ventajas y desventajas al utilizar constructores .- Tener el mismo nombre de la clase que inicializa***

***Ventajas:***- Puede definirse inline o fuera de la declaración de la clase  
- No devuelve valores  
- Puede admitir parámetros como cualquier otra función  
- Puede existir más de un constructor, e incluso no existir

***Desventajas:***

Los constructores tienen el mismo nombre que la clase, no retornan ningún valor y no pueden ser heredados. Además deben ser públicos, no tendría ningún sentido declarar un constructor como privado, ya que siempre se usan desde el exterior de la clase, ni tampoco como protegido, ya que no puede ser heredado.

***Ejemplo:***

#include <iostream.h>

class pareja

{

public:

//Constructor

pareja(int a2, int b2);

//Funciones miembros de la clase "pareja"

void Lee(int &a2, int &b2);

void Guardar(int a2, int b2);

private:

//Datos miembros de la clase "pareja"

int a, b;

}

pareja::pareja(int a2, int b2)

{

a = a2;

b = b2;

}

void pareja::Lee(int &a2, int &b2)

{

a2 = a;

b2 = b;

}

void pareja::Guarda(int a2, int b2)

{

a = a2;

b = b2;

}

int main(int argc, char \*argv[])

{

pareja par1(12, 32);

int x, y;

par1.Lee(x, y);

cout << "Valor de par1.a: " << x <<endl;

cout << "Valor de par2.b: " << x <<endl;

return 0;

}